LA-UR-21-22045
Approved for public release; distribution is unlimited.

Title: Eject, crash, or survive: Using machine learning to predict orbital instability of exoplanetary systems

Author(s): Smullen, Rachel Ann
Ayyalapu, Neha

Intended for: Mercer Science and Engineering Fair

Issued: 2021-03-01

EJECT, CRASH, OR SURVIVE: USING MACHINE LEARNING TO PREDICT ORBITAL INSTABILITY OF EXOPLANETARY SYSTEMS

Neha Ayyalapu
West Windsor-Plainsboro High School South
Senior Division, Mercer Science and Engineering Fair
Supervised by Rachel Smullen, Los Alamos National Laboratory

ABSTRACT

Astronomers throughout history, including titans like Kepler and Newton, have tackled planetary dynamics and orbital instability. Despite strides taken in research, understanding the evolution of planetary orbits remains an intricate, computationally expensive, and analytically unsolved problem. I apply machine learning classification methods to numerical simulations of planetary systems in order to predict the long-term fate of the planet—whether the planet remains in a stable orbit or not. My method uses the first 41.1 years (≤ 500 orbits) of data from a planet's simulation to calculate 17 dynamically-motivated metrics; I trained my classifier on these features to predict a planet's stability after 10^7 years. At 84.33%, my classifier was comparable in accuracy to pre-existing literature, despite using significantly less computational power than most other methods. In my research, I found that the standard deviation of eccentricity, mass ratios for neighboring planets, and semi-major axis ratio with the outer planet neighbor to be the most predictive features of instability. I propose reasons for the importance of these features, their role in planetary dynamics, as well as possible explanations for why some planets were misclassified. By understanding the important metrics of instability and reasons for misclassification, we can begin to understand more about system architectures, orbital motion and dynamics, and the formation and evolution of the exoplanetary systems. This is applicable in our own Solar System, and with exoplanet discovery missions such as *TESS*, this research becomes especially relevant in understanding the new exoplanetary systems we discover.

1 Introduction

In the last few decades, scientists have discovered more than 4,000 exoplanets—planets outside of the Solar System. Despite this success with missions such as *TESS* and *Kepler*, which found many planets by looking for small dips in light as the planets pass in front of their star, we have yet to find a system similar to our own. Understanding the architectures of real exoplanetary systems, and how that might relate to our own Solar System, requires us to understand the dynamics of diverse exoplanetary systems: their compositions, how they formed and evolved, and the gravitational influences planets exert on each other.

1.1 Orbits

Orbits in space (the paths that bodies travel as they move around one another under the influence of gravity) are characterized by six Keplerian orbital elements—semi-major axis, eccentricity, inclination, argument of pericenter, longitude of ascending node, and true anomaly—which describe the time-dependent position of one body around another. These orbital elements are described in Table 1.

The motions of bodies in the sky have been of interest since the ancient Greeks used circular orbits to describe the motion of the planets, the Moon, and the Sun. Hundreds of years (and many debates) after the Ptolemaic model, in the early 1600s, Johannes Kepler developed the modern theory of orbits and the three laws of planetary motion. Orbits are, arguably, even more critical to understand today than they were hundreds of years ago; they are the easiest property to measure for exoplanetary systems, and their potential to help understand the formations of systems make them ideal to use.

Table 1: Keplerian Orbital Elements and Other Orbital Quantities

Parameter	Symbol	Description
Mass	M	The mass of the star or planet
Semi-major axis	a	The average extent of the orbit
Eccentricity	e	The ellipticity of the orbit ($e = 0$ is a circle)
Inclination	i	The tilt of an orbit relative to a reference plane
Argument of pericenter	ω	The location of the closest approach (pericenter) of an orbit, relative to the point it crosses the reference plane
Longitude of ascending node	Ω	The angle between the reference direction and the point at which the orbit crosses from under the reference plane to over it
True anomaly	f	The position of the planet at the current time
Pericenter		The distance of the closest approach of an orbit
Jacobi distance		The distance of the body from the center of mass of all interior bodies

1.2 Dynamics

In the case of planetary systems, where multiple bodies have mass, the orbits of planets and stars evolve over time due to time-changing gravitational attraction. Typically, planetary systems can exist without substantial change for billions of years (such as in our Solar System). But, if the gravitational force between two planets becomes large (when two planets are too close together in physical space), the orbits can change substantially, which is termed as instability.

This instability can manifest in a few ways. It can cause planets to eject from the system: two or more planets exchange energy and/or angular momentum, and one planet, typically the lowest mass planet, gains enough energy to escape the system. Instability can also cause collisions between multiple planets or a planet and the star. As a result of orbital instability, planets are often lost from the system until the remaining planets can remain stable; their orbits will continue to slightly evolve, but they will not come in close contact or interfere with each other.

For two planet systems, there is a Hill criterion for dynamic stability (Gladman, 1993). If there are two planets, with masses m_1 and m_2 starting on circular orbits around a central star where $M_* > m_1 + m_2$, the system will remain stable if:

$$\Delta a = a_2 - a_1 > 2\sqrt{3}R_{H,m} \tag{1}$$

a_1 and a_2 are respective semi-major axes and $R_{H,m}$ is defined as the following:

$$R_{H,m} = \left(\frac{m_1 + m_2}{3M_*}\right)^{1/3} \frac{a_1 + a_2}{2} \tag{2}$$

Though this mutual Hill radius (MHR) criterion is only applicable in this idealized two planet system, it can still provide insight into planetary interactions and is a widely used measure of spacing for understanding planetary dynamics. The dimensionless dynamical separation is defined as the difference in semi-major axis divided by the mutual Hill radius.

$$\Delta = \frac{a_2 - a_1}{R_{H,m}} \tag{3}$$

For systems with more than two planets, instabilities can occur between adjacent planets beyond the two-planet MHR limit, and the time to instability has been shown to be a function of the global system spacing (Chambers et al., 1996).

1.3 Exoplanets and machine learning

The questions of orbital dynamics have been asked since Isaac Newton quantified the laws of gravity: how do systems organize themselves, how do their orbits evolve, what constitutes the laws of planetary motion?

However, one of the central challenges of studying this rises from the computationally expensive nature of running long-term gravitational simulations of planetary systems. For example, a 10^9 orbit integration using a timestep of 3.5% of the innermost planet's orbital period takes roughly 7 CPU hours (Rein & Tamayo, 2015). As such, there have been many undertakings to more quickly predict orbital instability from suites of N-body simulations, increasingly with the use of machine learning techniques. Related literature in the field explores classifications using initial conditions (e.g., Tamayo et al., 2020) and short-term integrations (Tamayo et al., 2016; Lam & Kipping, 2018) of systems of similar planets to predict the stability over millions of years in compact systems of two to five planets.

I apply machine learning to astrophysics in a similar way to explore system architectures and dynamics and to understand what drives orbital instability in diverse ten-planet systems. In this paper, I use characteristics of planetary orbits derived from very short numerical integrations to predict planetary instability.

2 Data and methods

In this section I describe the data I used in my research, as well as the methods I used to develop my machine learning model.

2.1 Simulations

The simulations used in this research were originally created in Smullen et al. (2016), which compared the evolution of planetary systems around single and binary stars. The N-body integrations were created using a Gauss-Radau variable timestep integrator in the MERCURY integrator package, which was used because of its ability to resolve interactions between planets effectively. Each simulation is run to an end time of 10 Myr, with instantaneous orbits output roughly every 13.69 years. The simulations cover 100 distinct systems, each with ten planets randomly drawn from the Mordasini planet population (Mordasini et al., 2009a,b) and a central star of one solar mass.

Approximately, the Mordasini population has a distribution of semi-major axis values from 0 to 100 AU (with a peak around 40 AU), eccentricity values from 0 to 0.3 with a peak around $e = 0.1$, planet masses between 0 and 10^4 Earth masses with peaks around 2, 20, and 1000 Earth masses (which correspond to about an Earth, a Neptune, and slightly larger than a Jupiter), and inclinations from $0°$ to $20°$ peaking around $4°$. The comprehensive set of conditions allows us to gain insight on exoplanetary systems with diverse architectures. These wider sets of parameters also allow us to study orbital dynamics in systems that more closely resemble observed exoplanets than most other research, which tend to have more less diverse system architectures.

Figure 1 shows the breakdown of planet fates in the Mordasini population (100 systems, totalling 1000 planets). Just under 30% of planets survive to 10 Myr, meaning that any system finishes with an average of ~ 3 planets. Nearly half of all planets are ejected from their systems. Collisions are quite rare, though planet-planet collisions are almost twice as common as planet-central body collisions.

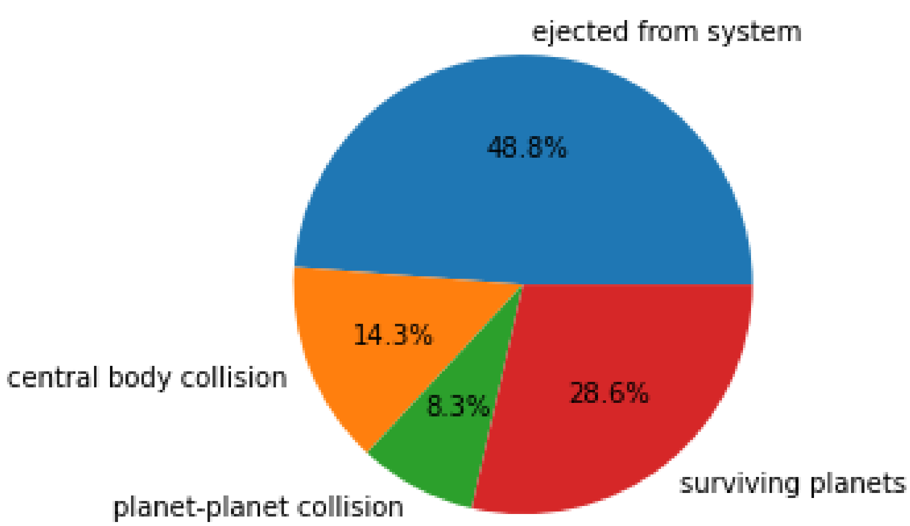

Figure 1: This pie chart shows the breakdown of the four possible planet fates (ejection, planet-planet collision, planet-central body collision, stable to 10 Myr) in the Mordasini population of 1000 exoplanets.

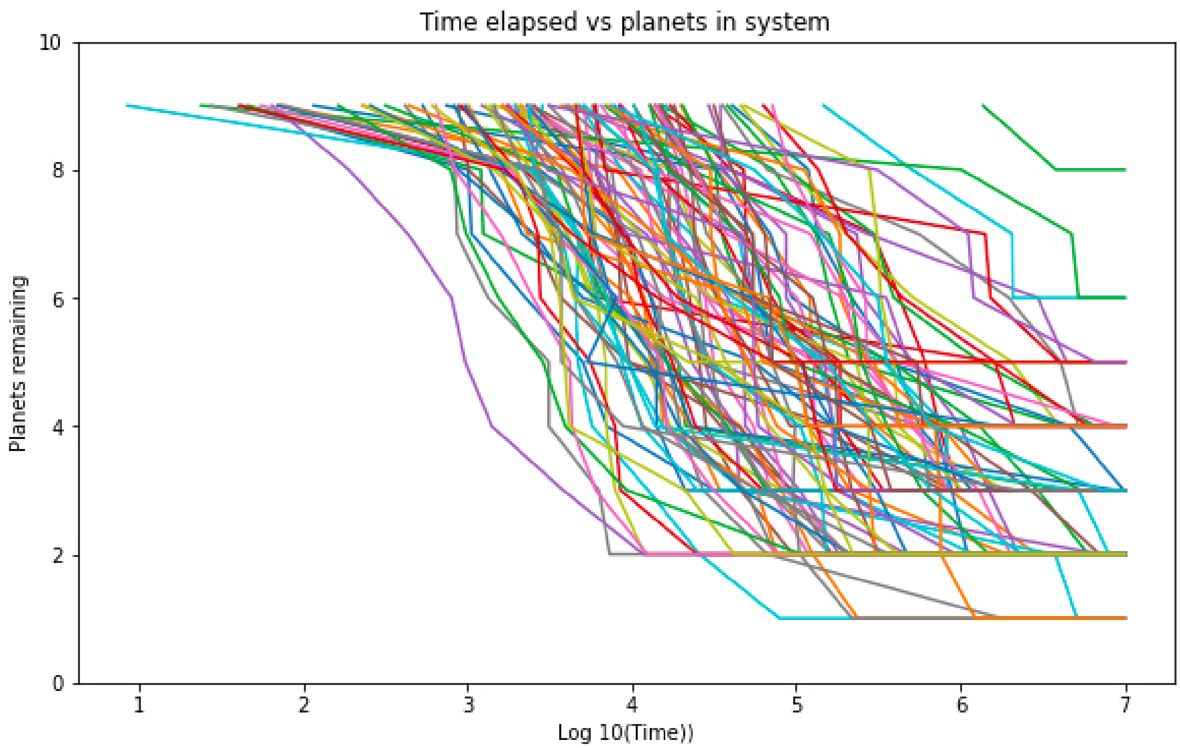

Figure 2: This graph shows the multiplicity of each of the 100 systems in the Mordasini population as a function of time. The wide variety of time-variable multiplicities show the diversity of instabilities in the Mordasini data.

2.2 Data processing

The methods used later in this work require features (measured properties of the thing one wants to classify) upon which to characterize planetary stability. To that end, this section summarizes my attempts to process the simulation data and translate the time-dependent orbital elements to features that might indicate orbital instability.

2.2.1 Time averaging

Current literature uses averaged quantities from short simulations as features, so one research goal was to investigate the shortest simulation I could use, (thereby minimizing computational efforts) while still being able to accurately predict the fate of each planet.

Using time averaging when exploring the derivations of values would enable me to look at the overall change of orbital conditions (initial and final conditions, minimum and maximum, standard deviation, and mean), which I expect could be significant in predicting orbital stability. As such, I tested a wide range of time averages, including a single timestep (at 13.7 years), 3 timesteps (27.4 years), 5 timesteps (54.8 years), 10 timesteps (123.2 years), 20 timesteps (260.1 years), 100 timesteps (1355.3 years), and 7306 timesteps ($100,005.5$ years).

Time averaging would also significantly reduce the amount of data I would eventually give to the machine learning classifier. Using too many variables can make our data more convoluted and difficult to interpret; it can also cause accuracy to decrease because of overfitting, which is when a classifier can predict one set of data so well that it impacts the performance of the model on new data.

2.2.2 Dynamical features

The simulations record six time-dependent orbital elements for each individual planet. On their own, these quantities do not provide any obvious information about the interactions between planets. So to create meaningful features to use in a machine learning classifier, it is useful to compute dynamically-relevant quantities for each planet in the time domain.

Because orbital evolution and instability is driven by interactions with nearby planets, quantities relating to mass and dynamical spacing may prove significant in the classifier (and have been shown to be indicators of instability in other works). I initially gathered data for mass, semi-major axis, Jacobi distance, and then I derived mutual Hill radius (MHR), mass ratio, and semi-major axis ratio with the neighboring planets.

In these simulations, planets are ordered by initial distance from the star and numbered from 1 to 10 (1 being the innermost planet). It has been shown by Cranmer et al. (2021) that a system's global stability could be estimated from the stability of sequential 3-planet component systems. So, on average, the two neighboring planets should exert the greatest influence on the dynamics of a given planet. For this reason, since MHR, mass ratios, and semi-major axis ratios involve values from interacting planets, I used the initial neighboring planets. For example, the neighbors of planet 5 were considered planet 4 and planet 6 for computational averaging. For the inner and outermost planets, I used the two next planets possible (e.g. planet 1's neighbors were computed as planets 2 and 3).

After all data processing, I computed features for each planet including: the initial value, final value, minimum, maximum, mean, and standard deviation of orbital quantities eccentricity, pericenter, Jacobi distance, and spacing in mutual Hill radii, mass ratio, and semi-major axis ratio for the neighboring planets.

2.3 Machine learning

Machine learning classification is a family of methods that sort objects into different classes (categories) based on combinations of their features (the data attributed to each object). Broadly, machine learning is based on the idea that systems can learn from data, identify patterns, and make classifications on new data using these patterns.

In general, after the initial steps of gathering and processing the data, one must select a classification method: the algorithm that analyzes this input data to make predictions. To create a machine learning model that can make predictions on data, one should use a training-testing scheme. With this, the classifier is *trained* on a fraction of the data set so that it learns the features of specific classes. It is then *tested* on the remaining data that it has not seen to estimate how well the model fits. When we first start the training process, an arbitrary 'line' is drawn through the multiple dimensions of the feature data. As training progresses, this line moves closer to the correct separation of the classes. Testing allows us to evaluate how accurate the trained classifier is, using data the classifier has never seen before (hold out). It is a metric of how well the model performs in the 'real world'. A standard testing-training split is 70% training data and 30% testing data.

Following testing and training is the process of refining hyperparameters, which are the variables that tell the classifier how to behave; changing hyperparameters can help make small improvements to the accuracy of the classifier. The grid search is an approach to hyperparameter tuning that methodically builds and evaluates models with every possible combination of the hyperparameters. A grid search can configure the optimal parameters for a given model. Following refining and re-testing the classifier, the model can now predict which class new data will belong to.

For this project in particular, in which I want to classify stable and unstable planets, I selected a gradient boosting random forest classifier, which is a supervised machine learning algorithm. As such, it first trains on data where it knows the correct classes (stable or unstable), and can then be used to predict the classes of new data. A random forest classifier creates an ensemble of decision trees, a flowchart-like structure with yes/no nodes representing specific features in the data. It then averages the results of each tree to determine the classification. I selected the gradient boosting random forest classifier because it has been used with high accuracy in related research (Tamayo et al., 2016; Smullen & Volk, 2020), and it also allows us to explore the importance of different quantities in the classification by determining relative feature importance.

In this research, a stable planet is defined as one that remains in the simulation for the entire 10 Myr. Though it can still undergo interactions with other planets, it is never further than 1000 AU from the star. An unstable planet is one that does not make it to the end of the simulation; it either suffers a planet-planet collision, planet-central body collision, or (most commonly) is ejected from the system.

Using the `scikit-learn` package (Pedregosa et al., 2011), I trained and tested a basic classifier for each of the datasets I created as described in Section 2.2.2, with the accuracies shown in Table 2.2.1, and plotted as a function of time in Figure 3.

The data averaged over 10^5 years showed the highest accuracy, but the long integration time is computationally intensive to simulate and average, and a large number of planets have already encountered instability by this point (see Figure 2). Therefore, averaging over three or five timesteps seemed like the best choice. As there was no functional difference between the two, I use the dataset that averages over three timesteps for the rest of the work presented in this paper.

Table 2: Classifier Accuracy as a Function of Time Averaging

Years	Number of timesteps	Accuracy
13.7	1 (skipped 0th)	82.33%
27.4	3	84.66%
54.8	5	84.66%
123.2	10	83%
260.1	20	84%
1355.3	100	86%
100005.5	7306	88%

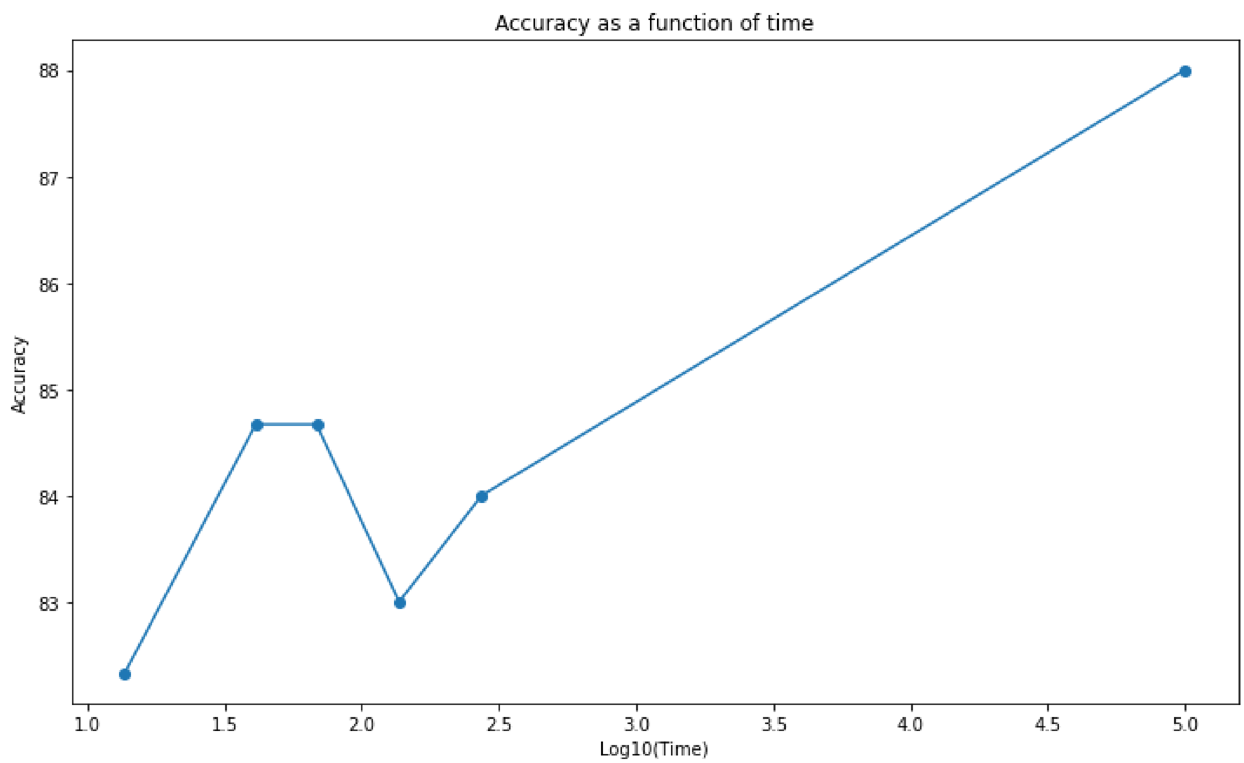

Figure 3: This graph uses the data from Table 2.2.1 to plot the classifiers' accuracies as a function of logarithmic-scale time averaging. Looking at accuracy relative to timescale is important in balancing computational efficiency and overall accuracy.

Machine learning classifiers can be subject to bias and overfitting, so to ensure I was using the smallest set of features that give a complete and accurate classification solution, I observed accuracy changing with different sets of features. Because I had calculated initial, final, minimum, maximum, mean, and standard deviation for each of the nine values I was testing with, there were significantly more features than necessary or important, and it clouded interpretability of the feature set.

As such, I elected not to use pericenter or Jacobi distance values, to completely eliminate the possibility of bias since these both have planetary population dependent physical units whereas all other features are dimensionless or scaled. Because mass ratios do not change over time as long as there are no collisions, I dropped all but one value for those variables. I tested with various combinations of the remaining features, and chose to train my classifier on the features in Table 3.

To increase the accuracy of my classifier, I then refined the hyperparameters, which are the variables that tell the classifier how to behave. Changing hyperparameters can help make small improvements to the accuracy of the classifier. Naturally, the ideal classifier would maximize the number of correctly classified planets with a high probability of belonging to the correct class and minimize the number of high-probability misclassified planets while also keeping the

Table 3: Features Used In Classification

Name	Shorthand
Initial eccentricity	initial e
Mean eccentricity	mean e
Standard deviation of eccentricity	sd e
Mass ratio with inner planet	mass1_ratio
Mass ratio with outer planet	mass2_ratio
Initial semi-major axis ratio with inner planet	initial a1_ratio
Mean semi-major axis ratio with inner planet	mean a1_ratio
Standard deviation of semi-major axis ratio with inner planet	sd a1_ratio
Initial semi-major axis ratio with outer planet	initial a2_ratio
Mean semi-major axis ratio with outer planet	mean a2_ratio
Standard deviation of semi-major axis ratio with outer planet	sd a2_ratio
Initial mutual Hill radius with inner planet	initial mhr1
Mean mutual Hill radius with inner planet	mean mhr1
Standard deviation of mutual Hill radius with inner planet	sd mhr1
Initial mutual Hill radius with outer planet	initial mhr2
Mean mutual Hill radius with outer planet	mean mhr2
Standard deviation of mutual Hill radius with outer planet	sd mhr2

overall classifier accuracy as high as possible. When refining, I looked primarily at the distribution of class probabilities as the accuracy was always within a small margin of values (84% to 86%). Using the grid search cross validation method, I refined the classifier to arrive at the optimal solution to maximize accuracy and minimize high probabilities in misclassified objects. The best fitting classifier had a learning rate of 0.09, a maximum tree depth of 4, a maximum number of features scaled by the base 2 logarithm of the number of samples, and 79 estimators; this classifier achieved an accuracy of 84.33%. I chose these hyperparameters because they minimized misclassified planets with probabilities above 90% and 95% relative to the other classifiers. I had identified a reasonable target to be $< 30\%$ of planets misclassified above 90%, and this classifier met that goal.

3 Results and analysis

In this section, I describe the findings regarding my analysis of the classifier's feature importances, the correlations between significant features, possible reasons why planets might be misclassified, and class membership probabilities.

3.1 Feature importance

With a trained and tested classifier, I sought to examine the importance of individual features and their correlation, if any, to orbital dynamics. A feature's importance is a fraction representing the percentage of the classification decision that relies on the specific feature in question. The ten most important features, ranked from highest to lowest importance, are shown in Figure 4.

Eccentricity's standard deviation does not come as much of a surprise for being the most important feature. Changes in eccentricity—when a planet's orbit is becoming more or less elliptical—is an indication of orbital evolution, which is often a precursor to orbital instability as an ejection or collision. The next most important features are the mass ratios for the two neighboring planets. Mass ratios can help estimate the strength of gravitational interactions between planets, so these quantities having a high importance also makes sense.

Also unsurprisingly, there are several semi-major axis ratios and mutual Hill radius values that are important to the classifier. Like eccentricity, changing semi-major axis values may also be an indicator of instability, with planets' orbits growing or shrinking as they interact with a more dominant neighbor. Additionally, the mutual Hill radii serve as a metric of dynamical spacing because they depend on both mass and distance; it has previously been shown to correlate to instability (e.g., Obertas et al., 2017).

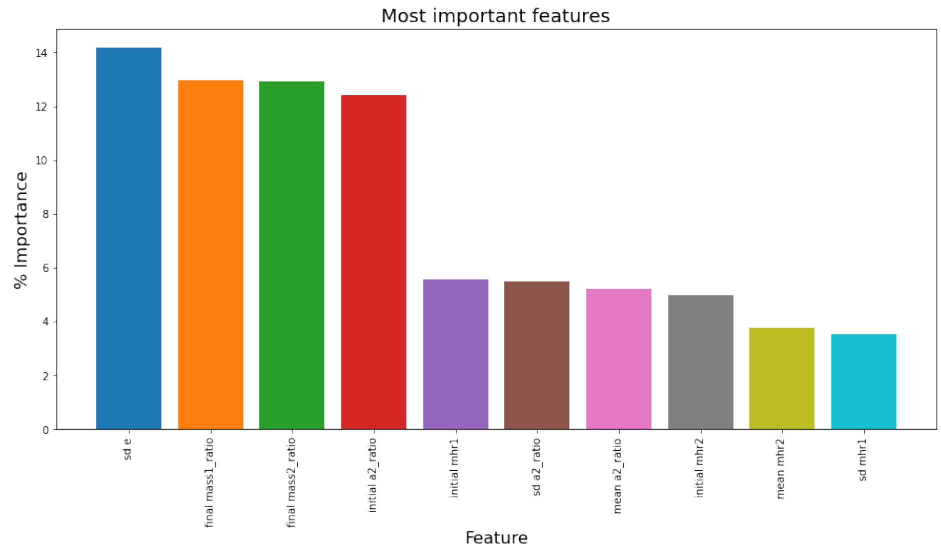

Figure 4: This bar graph shows the percent importance of the ten most important features used in my classifier. Isolating the importance of specific features allows us to better understand what drives orbital instability in these systems.

3.2 Feature correlations

The ordering of feature importance poses questions about correlations between features for correctly classified planets and misclassified planets. Figures 5 and 6 show one such example of a correlation in which I plot the standard deviation of eccentricity against the mass ratio with neighboring planets.

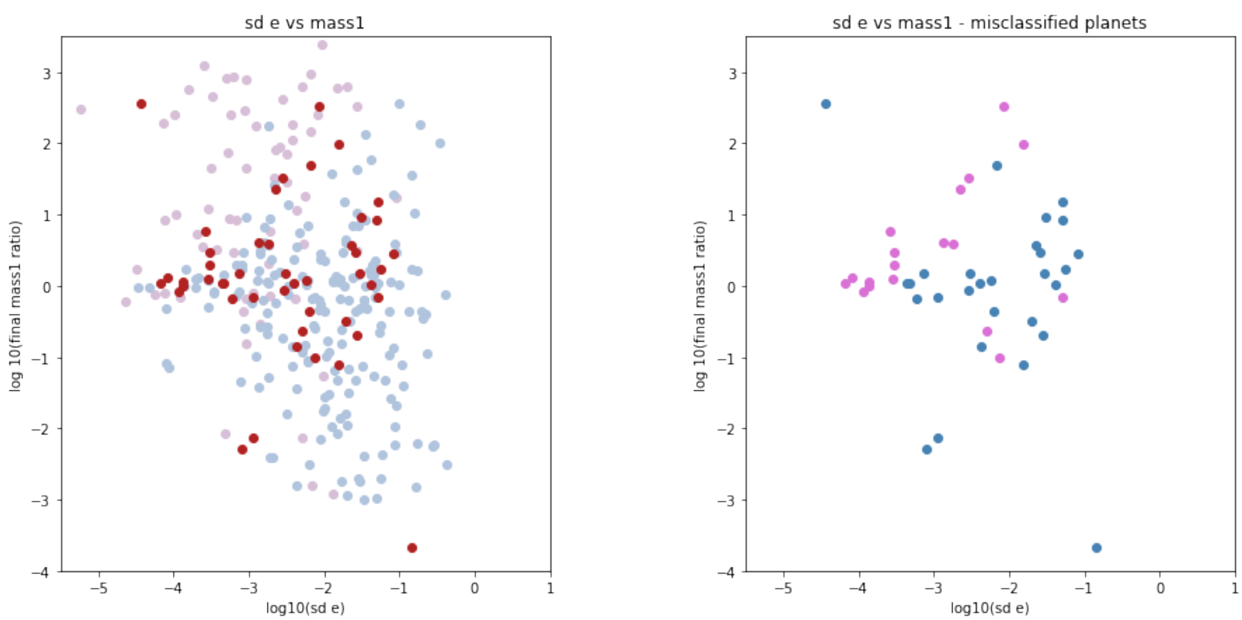

Figure 5: These two plots graph the two most important features against each other (sd e and mass_1 ratio). The left plot shows all planets in the testing set, and is colored by unstable (blue), stable (purple), and misclassified (red) planets. The right plot shows misclassified planets only; planets that were predicted stable and are actually unstable are in purple, while planets predicted unstable and are truly stable are blue. Observing such graphs may provide insight to the nature of misclassified planets, and why they may have been incorrectly predicted.

The stable planets are mostly clustered smaller values of sd e on the vertical axisat small values of sd e on the horizontal axis, which is expected as these planets should exhibit little orbital evolution over the short time span that the classifier

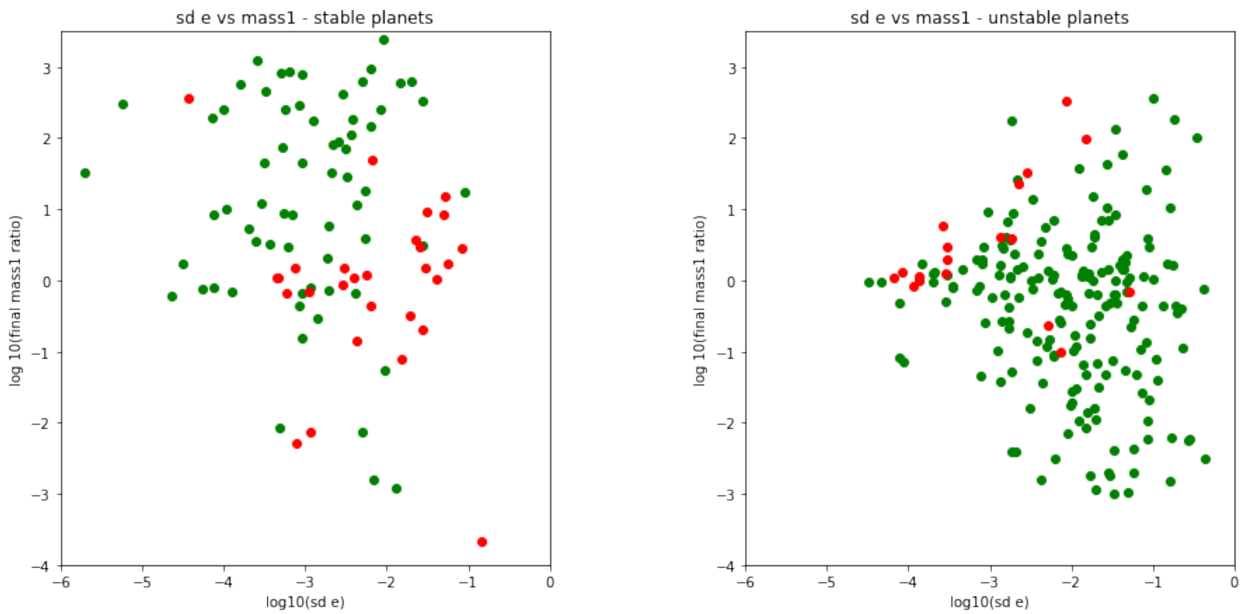

Figure 6: Like Figure 5, this figure graphs the two most important features against each other. The first plot has all points in the testing set for stable planets, and the second has unstable planets. In both, green points were correctly classified, while red ones were misclassified. The misclassified planets in both sets tend to lie on the edge of the cloud of points, perhaps leading to insight as to why they were misclassified.

is trained upon. Because eccentricity often changes due to the gravitational interactions with other planets, the trend of increased deviance in eccentricity for unstable planets is expected. Additionally, most of these that were misclassified—predicted unstable, but actually stable—either had a logarithmic mass1_ratio less than 0 or logarithmic sd e greater than -2 (or both). On the other hand, most of the correctly classified unstable planets are in the range mass1_ratio< 0, or have y-values greater than 0 coupled with x-values greater than -2. Nearly every single incorrectly classified unstable planet (predicted stable, actually unstable) was near or above 0 on the vertical axis.

Because a logarithmic mass1_ratio less than 0 implies that the planet's mass is less than its neighbors, these findings show the expected result that most unstable planets are those that are less massive than their neighbors. The gravitational influences these larger neighbors exert cause the planet to become unstable, which also corresponds to larger deviations of eccentricity in these planets.

The data also suggests there are some unstable planets equally or more massive than their neighbors. While these planets also tend to have higher eccentricity standard deviations, the higher mass ratios indicate there may be other dynamical influences on these planets. Upon further investigation, roughly 49% of these unstable planets with logarithmic mass_1 ratios ≥ 0 have logarithmic values less than 0 for mass2_ratio, meaning they are the less massive planet relative to their second neighbor. For the remaining planets, there may be other features (possibly semi-major axis ratios and mutual Hill radii) playing a role in these planets' fates.

3.3 Misclassified planets

To further explore the behavior of planets that were misclassified, I chose two examples of misclassified planets and plotted semi-major axis and eccentricity as functions of time for whole system.

One such graph is shown in Figure 7, which shows a planet that was misclassified as unstable despite the fact that it is actually stable. Though this planet is but an example, the relatively unchanging values of both eccentricity and semi-major axis in the time duration that classifier uses poses questions as to what might have led the classified to predict it unstable. Both semi-major axis and eccentricity seem relatively unchanging, so other features are likely contributing to the unstable classification.

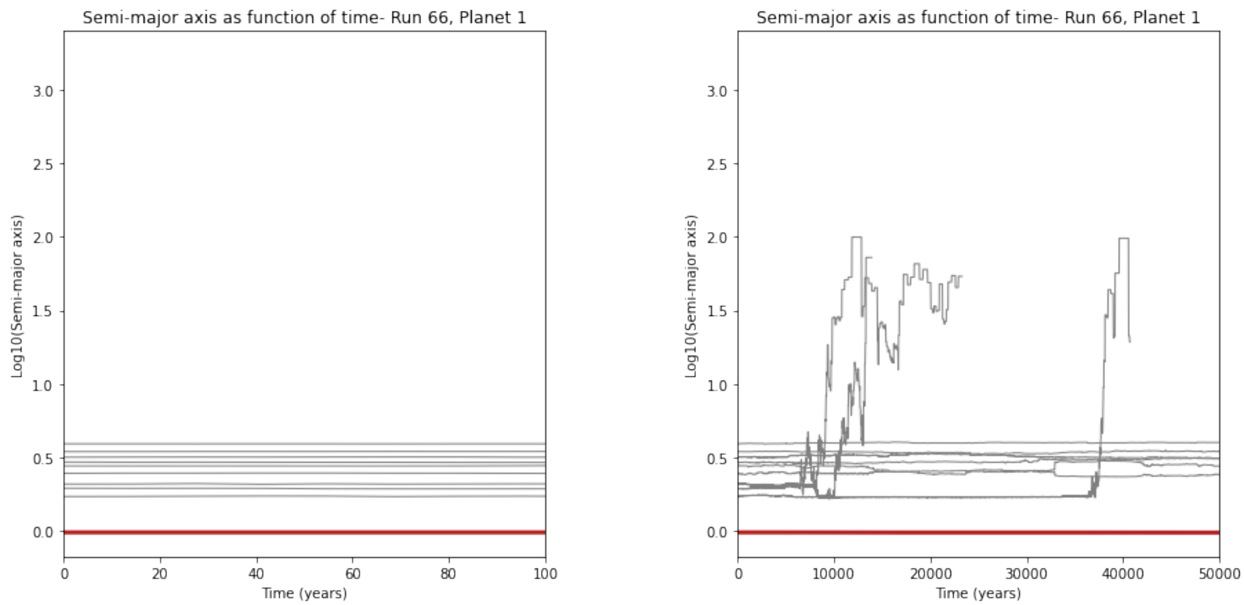

Figure 7: This is log-scale semi-major axis graphed as a function of time for a planetary system with a planet (in red) misclassified as unstable, in comparison to the other correctly classified (gray) planets in the system. The first plot shows the first few outputs of the simulation (100 years), whereas the second plot extends to 50000 years

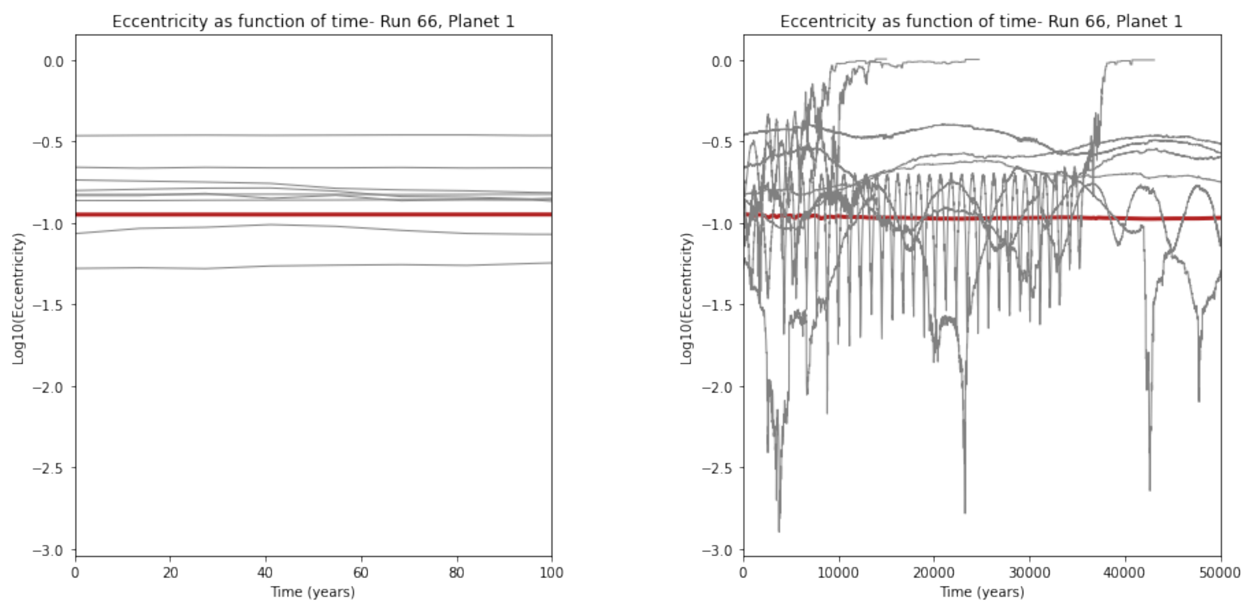

Figure 8: This plot follows the format of Figure 7, but graphs log-scale eccentricity as a function of time for the same planet

Because I had only used 41.1 years in my data (a very small period of time relative to the total simulation time) one would expect the typical misclassified planet to be classified as stable, and actually be unstable. With simulations integrated for such long periods of time, planets have millions of years for a single close encounter to result in instability which might not be apparent in the initial condidions. An example of this type of misclassification is depicted in Figures 9 and 10 (predicted stable, actually unstable).

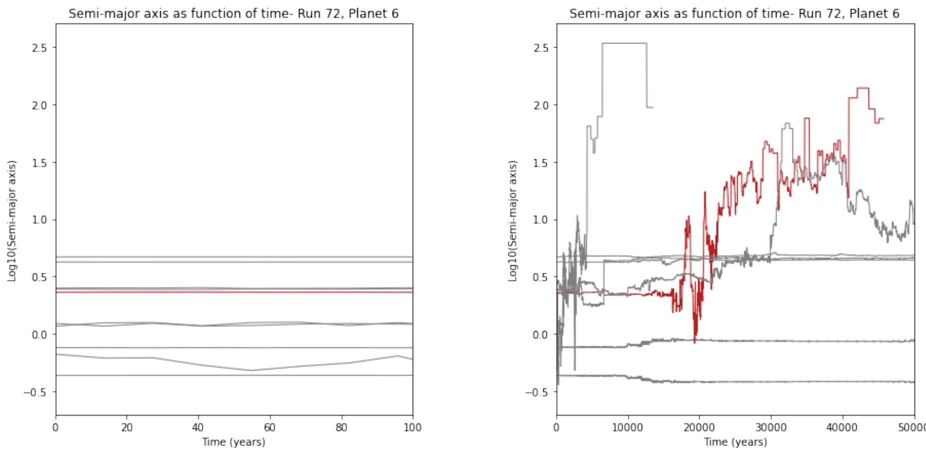

Figure 9: Following the format of Figure 7, this graph shows semi-major axis on a logarithmic scale over time for one system. However, this is a model of a planet misclassified as stable.

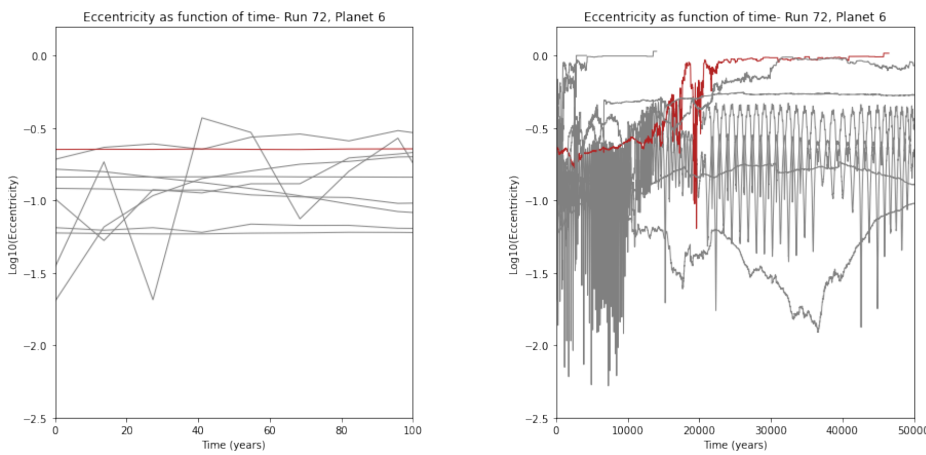

Figure 10: Like Figure 8, this shows logarithmic eccentricity as a function of time, using the same Mordasini system as in Figure 9 with the focal planet being misclassified as stable

Clearly, the semi-major axis and eccentricity are relatively unchanging during the time the classifier's data is used to make predictions. But, when viewed with the larger time frame, both values become irregular before the planet either ejects from the system. oopsie

3.4 Probability

Another thing to investigate is the probability that a certain planet belongs to either the unstable or stable class. The classifier returns a probability of class membership (in this case, the two classes are stable and unstable) where the sum of the probabilities for all classes is 100%. The predicted class is then chosen as the class with the highest probability. Figure 11 summarizes the classifier's predictions.

Planets that were predicted to be stable but actually unstable, like Figures 9 and 10, were the case for 19 of the misclassified planets, and the average probability of class membership for these classifications was 68.83%. One explanation for these types of misclassifications might be that the planets were initially stable, but nearby planets had orbital instability that impacted the orbits of these planets later in the simulation.

The other 28 misclassified planets were predicted to be unstable but were actually stable (like Figures 7 and 8). The average probability for these planets was 82.44%. One possible explanation for why stable planets might be misclassified as unstable is that some of the misclassified planets may have undergone early orbital stutters. The

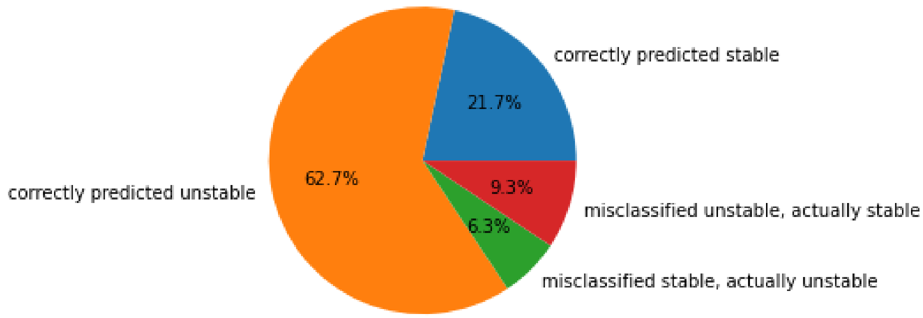

Figure 11: This plot shows the fractions of classification outcomes on the testing data— correctly predicted stable, correctly predicted unstable, incorrectly predicted stable, and incorrectly predicted unstable.

classifier may have encountered stronger indicators of instability in the features, such as a high sd e, despite the planets finding stable orbits later.

For correctly classified planets, the average probability was 86.01% for long-term stable planets and 89.62% for unstable ones. Though the $\sim 3\%$ point margin is nearly negligible, I offer some reasoning in Section 4.

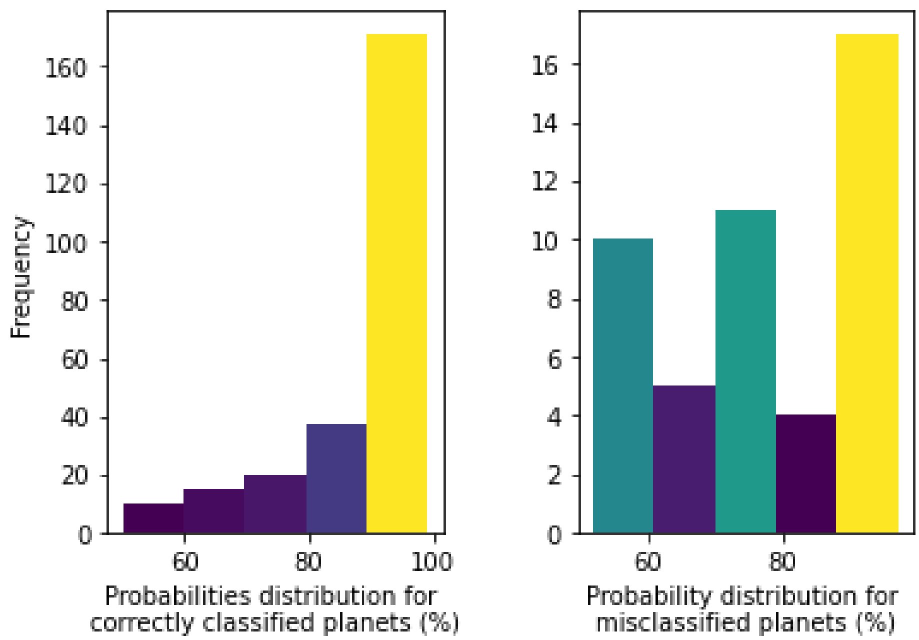

Figure 12: These histograms show the distribution of probabilities for correctly classified and misclassified planets using the testing set. It is important in evaluating the effectiveness of our machine learning algorithm because we want to maximize high probability in correctly classified planets and minimize high probability misclassifications.

More than just a high classifier accuracy, a machine learning method should, ideally, provide a high probability of class membership for correctly classified objects and a low probability of class membership for misclassified objects. The histograms in Figure 12 show the distribution of probabilities for correctly and incorrectly classified planets. As desired, the distribution for the correctly classified planets demonstrates a very strong curve, with a large majority of these planets having probabilities in the final bin (90% and above). The histogram for misclassified planets, however, shows a less than ideal circumstance. Though a majority of these planets are under the 75% mark, there is still a significant portion of them above 90%. This may be due to a fundamental limit in the time averaging: there will always be unpredictable cases when you only look at very short times.

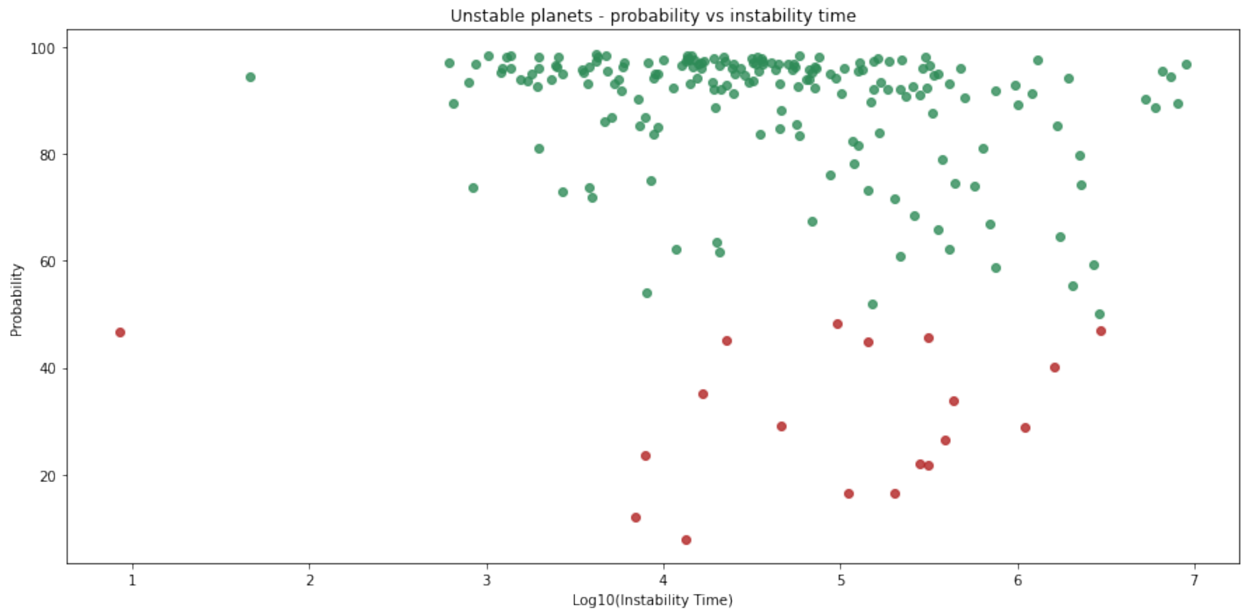

Figure 13: This plot shows the probability as a function of logarithmic instability time for all unstable planets in the testing set. The red points represent planets that were misclassified as stable, and the green represent correct predictions of unstable. Looking for a correlation between instability time and probability may help guide further improvements to the classifier.

The final thing I searched for was a correlation between accuracy and instability time, as I might expect that unstable planets with long instability times are more difficult to classify because they may exhibit behaviors of stable planets early on. There does not seem to be a significant trend in Figure 13. However, an important observation is that the later the instability time, the more spread apart the probabilities are as compared to earlier instability times, when the points are clustered at high class probabilities. This confirms that the classifier does struggle to provide a confident answer the later instability occurs in the planet.

4 Discussion and conclusion

In this research, I used data from N-body simulations of planetary systems to develop a machine learning classifier that tries to predict the long-term stability of individual planets. Using features derived from short-term integrations and a gradient boosting random forest machine learning classifier, I could classify systems' 10 Myr orbital stability in a computationally inexpensive way. After training, testing, and refining my classifier, I was able to achieve a total accuracy of 84.33%. The way that the classifier uses different features in the classification process helps me better understand the nature of these chaotic systems and what drives orbital instability. The most important findings of this work are as follows:

1. The standard deviation of eccentricity is, in my model, the most important feature to predict dynamical instability. Changes in eccentricity are an sign of orbital evolution, and a precursor to instability. My research also indicates that mass ratios of neighboring planets is an important factor in instability. I show that planets with neighbors of higher masses are significantly more likely to become unstable due to the gravitational influence these more massive planets exert.
2. The correlation I found between the classifier's most important features (sd e and mass1_ratio) is consistent with expectations. The relationship between these features also offered a hint for why planets were misclassified because the majority existed near the boundary of the stable/unstable distributions. or should be on the outer cloud for that classifications other points
3. This work suggests that discrepancies in the timescales of instability can cause misclassifications. Planets that are actually unstable but classified as stable may have late-time interactions with other planets that lead to

instability. Conversely, planets that are actually stable but classified as unstable may have undergone early orbital stutters before settling into a different, but stable, orbit.

4. While correctly classified planets have a high probability of class membership, there is a much wider spectrum of class membership probabilities for misclassified planets. This larger distribution may be because of the fundamental limit when time averaging, as there will always be unpredictable cases when only using data from a time as early as 41.1 years (≤ 500 orbits).

My classifier achieved a similar accuracy as other literature in the field. However, using numerical integrations of 41.1 years, my classifier uses significantly less data than other literature in the field (also making it less computationally exhaustive). However, this does lead to a few limitations.

First, the 10^7 year integration timescale is much shorter than the average lifespan of known planetary systems; exploring the evolution of these planetary systems for longer periods of time may change my results. Additionally, despite extensive work to improve and further refine my classifier, the inability to get accuracy above 88%, even when averaging up to 10^5 years of simulation data, suggests there may be some fundamental limit on the accuracy for this data, which could possibly be due to physics instead of the algorithm.

Also important to note is the limitations of the Mordasini planet population itself. This planet population is only one of many proposed initial distributions of planets, so I could further explore the performance of this method on different initial conditions.

In the future I could also measure other metrics of classifier quality, including accuracy, precision, and recall rates, which may guide improvements to my classifier and serve as useful comparators to models developed by other researchers. I could also adapt the classifier to predict the fate of the planet beyond just the binary classification of stable or unstable, meaning that my classifier could learn to predict ejections, planet-planet collisions, planet-central body collisions, and surviving planets. However, this goal may be limited by the somewhat unbalanced dataset, as only $\sim 28\%$ of the Mordasini planets are stable at the end of the simulation, and the very limited number of collisions. Machine learning algorithms are sensitive to the proportions of different classes, so with unbalanced data, they tend to favor the majority class, with the largest number of observations. Without enough data for each class, the classifier may fail to make accurate predictions and can be particularly problematic when predicting rare classes such as planet-planet collisions.

This work serves as another contribution to the growing field machine learning and computing techniques applied to orbital dynamics. Not only am I able to generate these large datasets with ease, but machine learning's predictive power allows us to better understand these chaotic systems in a faster and less computationally exhaustive way.

Besides the predictive power of my classifier, the search for exoplanets with projects such as *TESS* continues. Applying machine learning to astrophysics offers the potential to help us understand more about planet compositions, configurations, and system architectures more precisely and on a much larger scale than ever before. Further research in this realm will not only tell us more about planetary motion and dynamics, but it may lead to discoveries about the formation and evolution of these new systems we discover and our own Solar System.

References

Chambers J. E., Wetherill G., Boss A. P., 1996, Icarus, 119, 261

Cranmer M., Tamayo D., Rein H., Battaglia P., Hadden S., Armitage P. J., Ho S., Spergel D. N., 2021, arXiv:2101.04117 [astro-ph, stat]

Gladman B., 1993, Icarus, 106, 247

Lam C., Kipping D., 2018, Monthly Notices of the Royal Astronomical Society, 476, 5692

Mordasini C., Alibert Y., Benz W., 2009a, Astronomy and Astrophysics, 501, 1139

Mordasini C., Alibert Y., Benz W., Naef D., 2009b, Astronomy & Astrophysics, 501, 1161

Obertas A., Van Laerhoven C., Tamayo D., 2017, Icarus, 293, 52

Pedregosa F., et al., 2011, Journal of Machine Learning Research, 12, 2825

Rein H., Tamayo D., 2015, , 452, 376

Smullen R. A., Volk K., 2020, Monthly Notices of the Royal Astronomical Society, 497, 1391

Smullen R. A., Kratter K. M., Shannon A., 2016, Monthly Notices of the Royal Astronomical Society, 461, 1288

Tamayo D., et al., 2016, The Astrophysical Journal, 832, L22

Tamayo D., et al., 2020, Proceedings of the National Academy of Sciences, 117, 18194